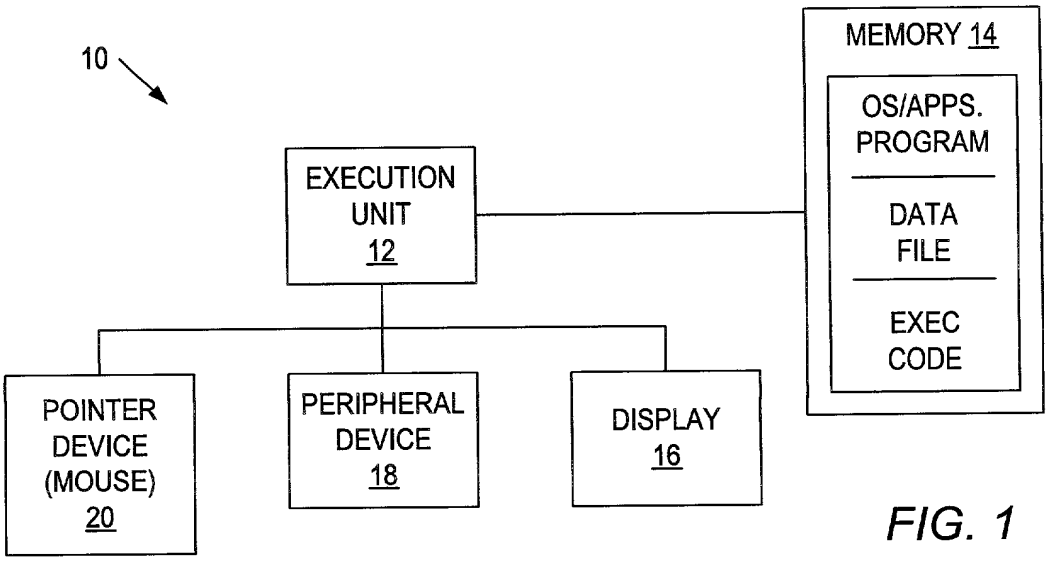
10
MEMORY 14
OS/APPS. PROGRAM
DATA FILE
EXEC CODE
EXECUTION UNIT 12
POINTER DEVICE (MOUSE) 20
PERIPHERAL DEVICE 18
DISPLAY 16

FIG. 1

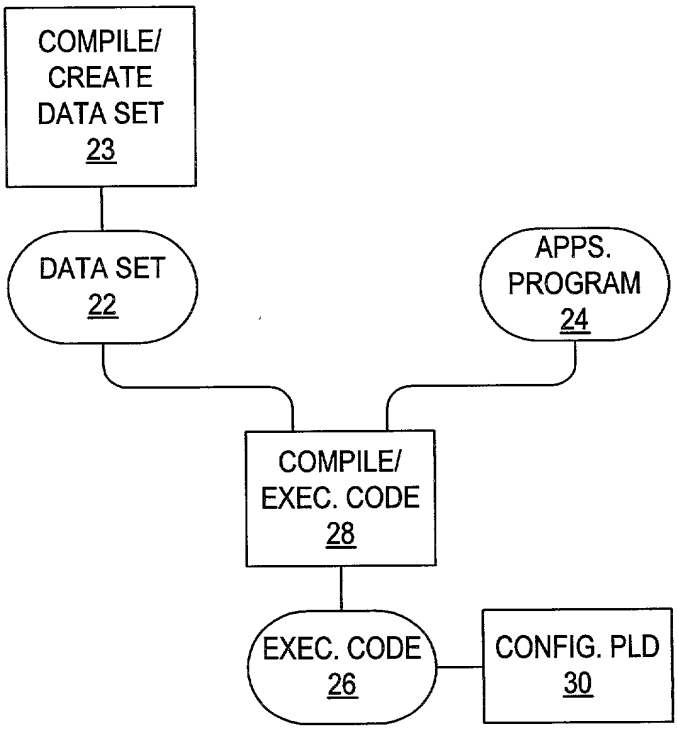
COMPILE/ CREATE DATA SET 23
DATA SET 22
APPS. PROGRAM 24
COMPILE/ EXEC. CODE 28
EXEC. CODE 26
CONFIG. PLD 30

FIG. 2

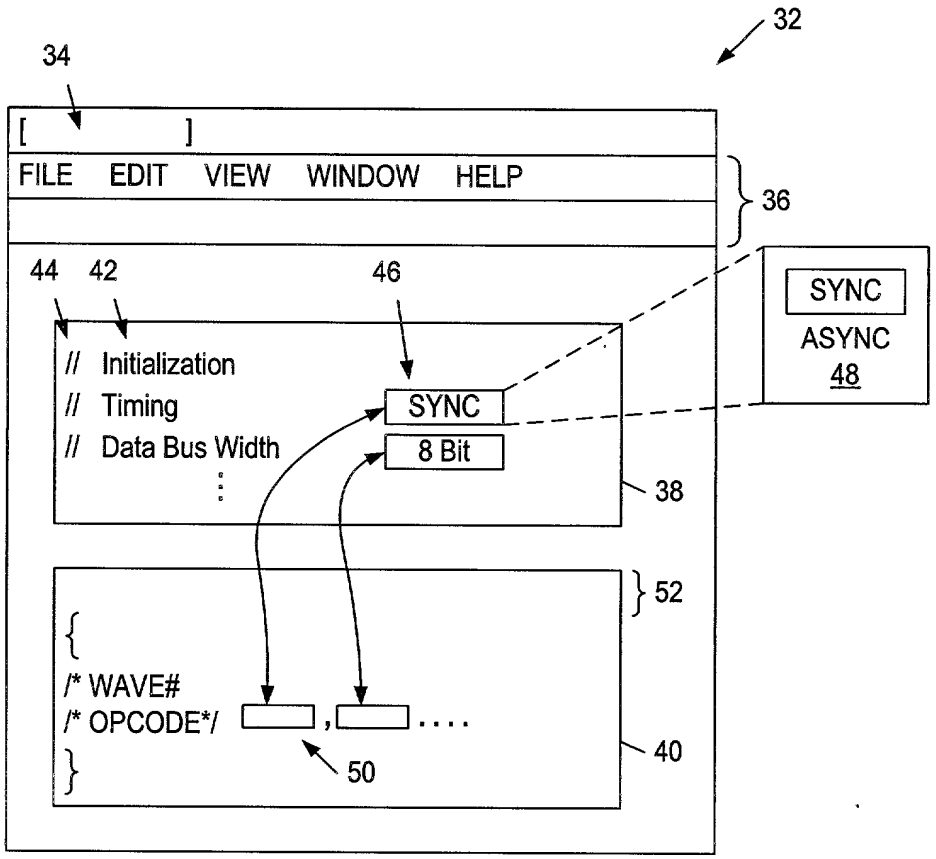
32
34
[ ]
FILE EDIT VIEW WINDOW HELP
36
44 42 46
// Initialization
// Timing
// Data Bus Width
SYNC
8 Bit
38
SYNC
ASYNC
48
52
{
/* WAVE#
/* OPCODE*/
}
50
40

FIG. 3

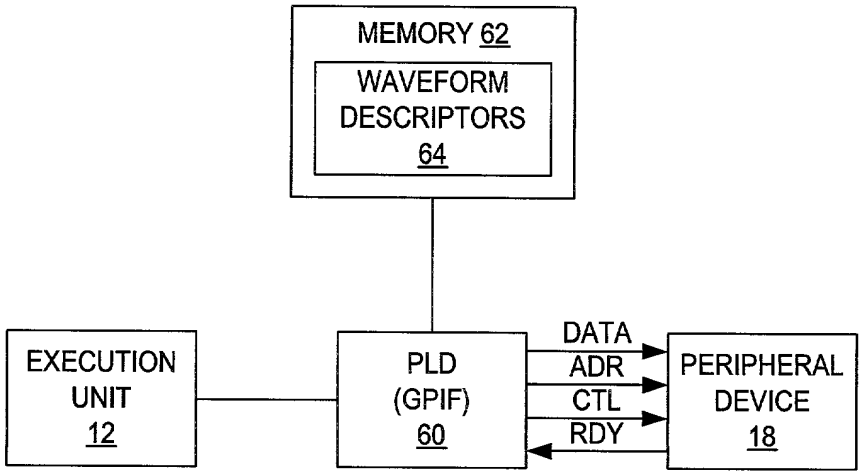
MEMORY 62
WAVEFORM DESCRIPTORS 64
EXECUTION UNIT 12
PLD (GPIF) 60
DATA
ADR
CTL
RDY
PERIPHERAL DEVICE 18

FIG. 4

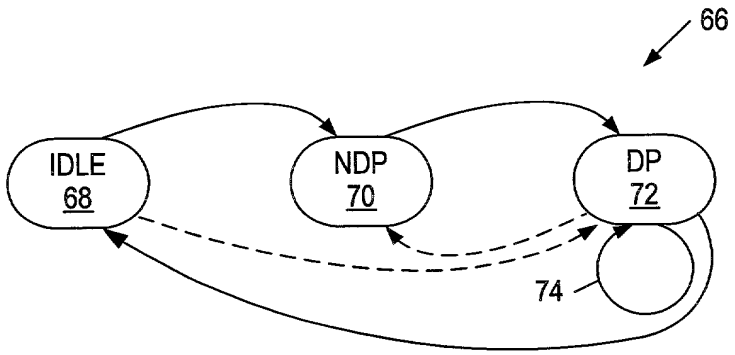
66
IDLE
68
NDP
70
DP
72
74

FIG. 5

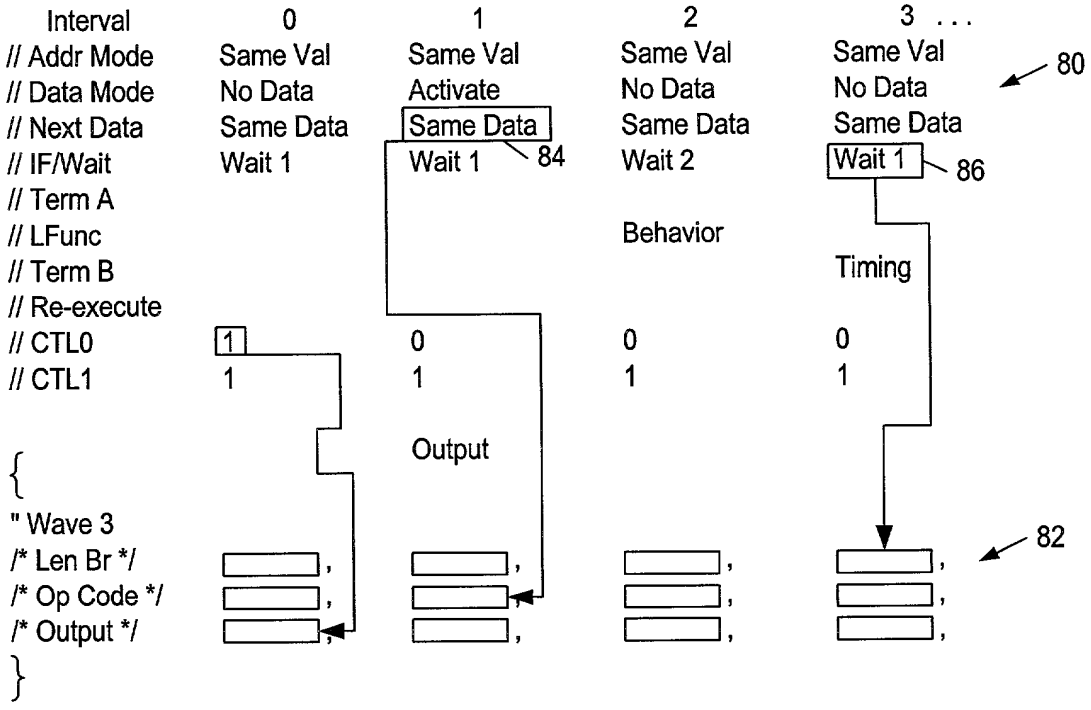
Interval
0
1
2
3 . . .
// Addr Mode
Same Val
Same Val
Same Val
Same Val
80
// Data Mode
No Data
Activate
No Data
No Data
// Next Data
Same Data
Same Data
Same Data
Same Data
// IF/Wait
Wait 1
Wait 1
84
Wait 2
Wait 1
86
// Term A
// LFunc
Behavior
// Term B
Timing
// Re-execute
// CTL0
1
0
0
0
// CTL1
1
1
1
1
Output
{
" Wave 3
/* Len Br */
/* Op Code */
/* Output */
82
}

FIG. 6

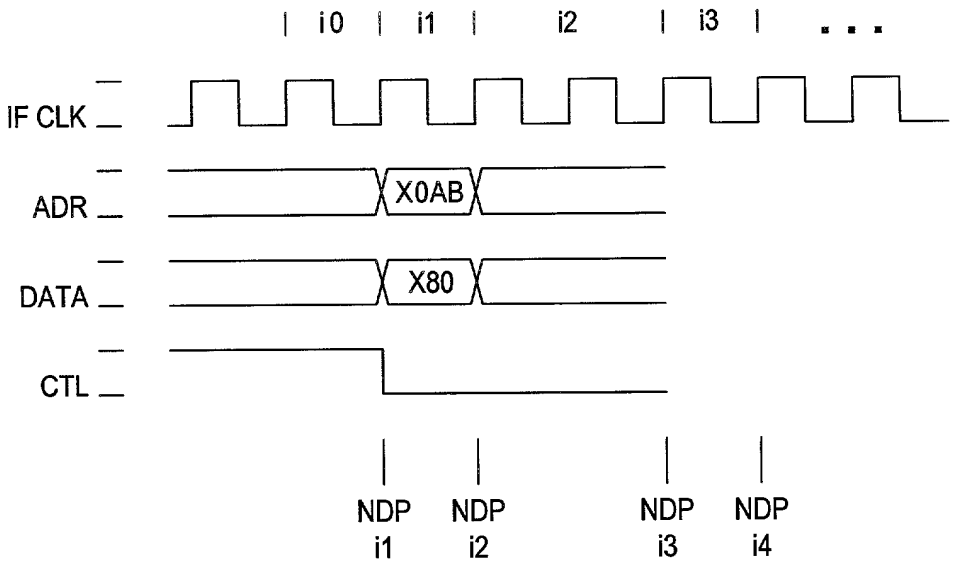
i0
i1
i2
i3
...
IF CLK
ADR
X0AB
DATA
X80
CTL
NDP i1
NDP i2
NDP i3
NDP i4

FIG. 7

```
#define PERIPHCS 0x00AB
#define AOKAY 0x80
#define BURSTMODE 0x0000
#define TRISTATE 0xFFFF
#define EVER ;;

// prototype
void GPIFInit ( void );

// Set Address GPIFADR [8:0] to PERIPHERAL
void Peripheral_SetAddress ( WORD gaddr )
{
 if ( gaddr < 512 )
  { // drive GPIF address bus w/gaddr
    GPIFADRH = gaddr >> 8;
    GPIFADRL = ( BYTE ) gaddr; // setup GPIF address
  }
  else
  { // tri-state GPIFADR [8:0] pins
    PORTCCFG = 0x00; // [7:0] as port I/O
    OEC = 0x00; // and as inputs
    PORTECFG & = 0x7F; // [8] as port I/O
    OEC & = 0x7F; // and as input
  }
}

// read single byte from PERIPHERAL, using GPIF
void Peripheral_SingleByteRead ( BYTE xdata *gdata )
{
  static BYTE g_data = 0x00

  while( ! ( GPIFTRIG & 0x80 ) ) // poll GPIFTRIG.7 Done bit
  {
      ;
  }

  // using register(s) in XDATA space, dummy read
  g_data = XGPIFSGLDATLX; // to trigger GPIF single byte read transaction

  while( ! ( GPIFTRIG & 0x80 ) ) // poll GPIFTRIG.7 Done bit
  {
      ;
  }

  // using register(s) in XDATA space, GPIF read byte from PERIPHERAL here
  *gdata = XGPIFSGLDATLNOX;
}
```

FIG. 8